Reliable Tailored-COTS via Independent Verification and Validation

Michael A. Beims
AverStar, Inc.
100 University Drive
Fairmont, WV, USA 26554-8818
mbeims@mail-fair.ivv.nasa.gov

James B. Dabney
AverStar, Inc.
1100 Hercules, Suite 300
Houston, TX, USA 77058
jim@averstar.com

Abstract

An important class of Commercial Off-The-Shelf (COTS) applications is the adaptation of an established COTS product to an operational environment for which it was not originally intended. This tailoring of the established product can provide the expected cost-reduction benefits associated with COTS and still meet system reliability requirements when augmented with an appropriate Independent Verification and Validation (IV&V) activity. We illustrate the tailored-COTS IV&V approach using the integration of a COTS Global Positioning System (GPS) receiver into the Space Shuttle onboard avionics system. The COTS GPS receiver chosen is a proven, reliable navigation aid that has been successfully integrated in numerous military aircraft, ranging from helicopters to jet fighters. However, integration of this COTS receiver into the Space Shuttle avionics system required many changes due to the different avionics hardware environment and the dramatically different flight environment. The key elements of the tailored-COTS IV&V approach are identification of unchanged but operationally affected code, development of automated code analysis tools, software scenario analysis, and exploitation of historical databases.

1 Introduction

Tailored-COTS is an important class of COTS applications in which proven off-the-shelf equipment is adapted to environments for which it was not originally intended. Tailored-COTS can be quite attractive economically, but it presents special challenges to Independent Verification and Validation (IV&V). The integration of the COTS GPS receiver into the Space Shuttle avionics system illustrates typical problems that must be overcome in a tailored-COTS program. The selected GPS receiver is a proven off-the-shelf product that has been successfully integrated into the avionics systems of numerous military aircraft. Changes required for integrating this COTS GPS receiver into the Space Shuttle avionics system include a new interface where a Space Shuttle-specific serial input/output (I/O) card replaced the Mil-Standard 1553 bus Serial I/O interface. Also, the orbital flight environment required significant changes to navigation and satellite vehicle acquisition and tracking algorithms designed for relatively (compared to the Space Shuttle) low speed and low altitude atmospheric flight.

As a result of these significant changes and the high criticality of the Shuttle navigation system, NASA's Independent Verification and Validation facility was tasked to perform IV&V on the Shuttle's modified COTS GPS receiver, specifically the embedded software in the receiver. This IV&V effort required the development of a new tailored-COTS IV&V process that has been very successful. This new IV&V process was based on IV&V techniques employed successfully on traditional mission-critical software development projects. The tailored-COTS environment presents significant new issues in resource allocation and verification and validation techniques.

The paper briefly describes the hardware and environmental differences between the COTS GPS receiver's environment and the Shuttle, and explains the unique issues posed by IV&V of tailored-COTS products. This paper also identifies several IV&V techniques that were successfully used during IV&V of the modified COTS GPS receiver's embedded software. Finally, it presents conclusions and suggests future improvements to the process.

2 Background

The Space Shuttle is a unique aerospace vehicle in that it must operate as a rocket (during launch and ascent), as a satellite (during orbit), and as an aircraft (during entry and landing). These distinctly different flight regimes each present different navigation problems. The current Shuttle navigation system uses star tracker and ground radar for on-orbit navigation and tactical air navigation (TACAN) and microwave scanning beam landing system (MSBLS) during entry and landing. TACAN is a ground-based military enroute navigation system that is being replaced by GPS on all United States military aircraft. Therefore, within a few years, it will be necessary for NASA to replace the Shuttle TACAN system with GPS or to maintain the TACAN ground stations at NASA expense.

The selected COTS GPS receiver was designed and tested for use in military aircraft ranging from helicopters to supersonic jet fighter aircraft. It has proven to be an extremely reliable aid to navigation. Since the selected off-the-shelf unit is a military GPS

Paper presented at the RTO IST Symposium on "Commercial Off-the-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS"", held in Brussels, Belgium, 3-5 April 2000, and published in RTO MP-48.

receiver, it is equipped with the necessary circuitry to allow it to use the precise positioning service (PPS). This PPS capability provides increased accuracy over typical civilian GPS receivers, and reduces vulnerability to radio interference. All of these attributes are desirable for a Space Shuttle navigation system. Since developing a completely new GPS receiver for the Shuttle would be prohibitively expensive and the COTS GPS receiver has these desirable attributes, it was selected as the basis of a GPS receiver for the Shuttle.

Although the selected COTS GPS receiver is a proven, reliable product, there are still many differences between a typical military GPS application and the Space Shuttle. These differences include both the avionics environment and the flight environment. We will discuss each next.

2.1 Avionics Environment

Previous applications of the selected COTS GPS receiver provided control and user interface to the receiver through a control display unit or through the Mil-Standard 1553 bus. An interface manager function in the receiver accommodates the different interfaces, including service specific (Army, Navy, Air Force) variations in the 1553 bus controls. The Space Shuttle uses a modulator/demodulator (MDM) serial I/O bus, which requires a new hardware interface in the receiver and also new interface software in the receiver. Additional interface software changes inside the receiver were needed to process Space Shuttle flight software unique antenna lever arm and attitude references.

2.2 Flight Environment

There are several differences between the Shuttle flight environment and military aircraft. These include vehicle speed, altitude, and flight attitude. Although the original motivation for installing GPS in the Shuttle was replacement of TACAN (available only during the landing phase), GPS is available during all Shuttle mission phases. So the Shuttle avionics system was modified to use GPS in all flight phases, including the launch and orbit phases in addition to the landing phase.

Speed is a difference in the flight environment as typical speeds for military aircraft range from zero in hovering helicopters to less than Mach 3 for jet fighter aircraft. This contrasts with the Space Shuttle, which on orbit operates at speeds of up to Mach 25. Furthermore, navigation calculations for military aircraft are typically performed using either rhumb line or great circle techniques. Except during the landing approach, the Shuttle must use ballistic propagation algorithms.

A second consequence of the Shuttle's high speed is that satellites are typically visible for a much shorter period, thus increasing the satellite selection workload. For an aircraft, a satellite is typically visible for approximately six hours as the satellite traverses from horizon to horizon. For the Space Shuttle, this visibility window is reduced to approximately 45 minutes. Satellite selection in the COTS GPS receiver requires constantly choosing from among all the visible GPS satellites the set of four satellites that provides the best navigation solution. This is a complex calculation, and since the high speed of the Shuttle requires more frequent satellite selection computations, the computational resources available for other tasks are reduced.

Altitude presents another difference in the flight environment, since military aircraft typically fly at altitudes of less than 20 kilometers while the Space Shuttle flies at altitudes in excess of 500 kilometers. An important consequence of the Space Shuttle's increased altitude is that at any moment, more satellites can be visible to the Space Shuttle than are visible to an aircraft in atmospheric flight. This increases the number of satellites that must be evaluated for inclusion in the navigation solution, further increasing computational workload. Additionally, on orbit, the Space Shuttle has line-of-sight visibility to GPS satellites up to 20 degrees below the local level plane, potentially changing parameters of the satellite selection algorithms.

A final flight environmental difference is vehicle attitude. Military aircraft, including jet fighters, spend most of the time in a heads-up attitude. Therefore, for military aircraft a single GPS antenna on an upper surface has unobstructed line-of-sight to a sufficient number of GPS satellites most of the time. The Space Shuttle, on the other hand, frequently orbits in a heads-down attitude for extended periods. Also, during entry, the Shuttle flies at a relatively high pitch attitude, which obstructs line-of-sight to a large portion of the sky. Consequently, the Shuttle must use two GPS antennas, one on an upper surface and one on a lower surface. Since the navigation algorithms determine position based on the location of the receiving antenna, it is necessary for the software to decide which antenna is receiving the signal from each satellite, a problem not faced by the COTS GPS receiver.

2.3 Similarities to Other Applications

The differences between avionics and flight environments just described are significant and extensive. However, most of the COTS GPS receiver hardware and software were compatible with the Shuttle environment. For example, the basic hardware characteristics such as packaging and power required no change. Much of the COTS GPS receiver's internal software also required no changes including the radio frequency control processing, including the internal receiver moding and control, and the geometric calculations to reduce geometric dilution of precision (GDOP). Other unchanged off-the-shelf functions of particular importance are the military performance accuracy and the security related processing in the receiver (Selective Availability, anti-spoofing, and anti-

jamming). As these unchanged characteristics far exceed the new and changed characteristics; it is reasonable to treat the Shuttle's modified COTS GPS receiver as tailored-COTS rather than as an entirely new product.

3 Approach

The tailored-COTS environment presents significant new issues in resource allocation and verification and validation techniques. Other researchers have documented similar modifications to their processes for COTS applications. These modifications include process changes running through the entire range of the Procurement, System Engineering and Integration activities and have been documented for United States military procurements. Software engineering processes must be tailored to incorporate new computing system standards and methodologies. Avionics System Engineering processes must evolve and adapt to dynamically changing COTS Non-Developmental Item product lines that incorporate emerging standards [11]. While the solutions provided are employing commercial standards and off-the-shelf products, a major role to be played by the integrating organization is to become the trusted subsystem integrator. The organization will put wrappers around the commercial technologies to meet the customers' needs [17].

The first consideration in any IV&V effort is to determine the optimum allocation of finite IV&V resources. This process is complicated in the case of tailored-COTS because it is neither necessary nor economically feasible to perform comprehensive IV&V of the entire software product.

The software in a tailored-COTS product can be partitioned into three classes: new or modified, not modified but affected operationally, and unaffected. The first class, new or modified, is easy to assess since it clearly merits IV&V and can be dealt with using standard IV&V methods. The third class, unaffected, is also easy to assess, as it clearly does not merit IV&V. But the second class, not modified but affected operationally, presents two problems: identification and verification. The focus of this paper is the development and application of methods for identifying and verifying software code of the second class.

3.1 Criticality Analysis and Risk Assessment

A fundamental step in any IV&V project is the allocation of the available technical staff resource. Both the number of analysts and the overall project schedule constrain the activity. Since the amount of potential IV&V work on any complex project exceeds the available resources, it is necessary to allocate the resources to achieve the greatest benefit. NASA's IV&V contractor on this modified COTS GPS receiver project, AverStar, Inc., employs a process known as Criticality Analysis and Risk Assessment (CARA) to guide this resource allocation [12].

3.1.1 CARA Overview

CARA is based on the notion that there are two key factors to consider in IV&V resource allocation: criticality and risk. Here, criticality is a measure of the consequences of an error in a particular software function. Risk is a measure of the likelihood of an error. Table 1 provides a synopsis of the CARA process.

Table 1 : Criticality Analysis and Risk Assessment Process

Phase	Step	Activity
Preparation	1	Establish CARA team including domain experts and IV&V process experts.
	2	Decompose the software system into critical functions. These should be functionally distinct and sufficiently small to permit analysis by a single individual.
Evaluation	3	Develop criticality and risk criteria, starting with the baseline CARA factors.
	4	Rate each critical function using the selected criteria and compute overall CARA scores.
IV&V Scoping	5	Set threshold levels to map an IV&V level (degree of scrutiny) to the CARA scores.
	6	Perform software size estimates using measures such as source lines of code or function points.
	7	Estimate IV&V effort required using the size estimates of Step 6 and IV&V levels of Step 5.
	8	Repeat Steps 5 and 7 as necessary such that a feasible work plan is achieved.

CARA is an iterative process. It is performed once at the outset of an IV&V project, then repeated periodically. This iteration is necessitated by several factors. For example, as the project progresses, the IV&V team gains greater insight, enabling refinement of the analysis. Also, the software requirements and design can evolve, changing both criticality and risk, and even introducing new critical functions.

3.1.2 Shuttle COTS GPS Receiver CARA

For the Shuttle's modified COTS GPS receiver project, an initial CARA was performed after the IV&V team reviewed all available documentation. This included requirements and design documentation for the baseline military COTS GPS receiver and proposed changes to the COTS GPS receiver's embedded software to adapt it to the Shuttle. The team also reviewed applicable changes to the Shuttle general-purpose computer flight software. Additionally, the team analyzed development flight test data and operational requirements.

The tailored-COTS nature of the COTS GPS receiver IV&V project changed the CARA process significantly. Added factors in assessing risk were necessary to properly attribute risk reduction due to the shelf life of the COTS code. So it was necessary to identify and consider separately the changed and new code and the code that was not changed (or at least not changed much). Other new considerations that affected both criticality and risk were the different operational environment and the availability of historical data.

The differences between the operational environment of the Shuttle and previous applications of the selected COTS GPS receiver affected both criticality and risk. For example, the more rapid change in the relative configuration of the satellite constellation could amplify the consequences of errors in satellite selection algorithms. The differences in operational environment also increased the risk of problems in satellite selection because the algorithms must operate more frequently and track a larger number of satellites.

Risk analysis was expanded to include assessment of the degree to which each unchanged (or little-changed) critical function interacted with new or extensively changed critical functions. This determination was based on analysis of the software requirements and design documentation as well as mission analysis.

Risk analysis was augmented via problem databases maintained by the manufacturer and the United States Department of Defense. The reasoning was that critical functions, which had historically experienced a larger number of programming and operational errors, were considered more likely to contain errors with respect to the new environment.

Prior to the initiation of the IV&V effort, NASA had flown a prototype modified COTS GPS receiver on several Shuttle missions. These flight experiments provided a wealth of data that the IV&V team analyzed to gain further insight to aid the CARA.

The initial CARA guided the detailed requirements analysis phase of the modified COTS GPS receiver IV&V project. Subsequent CARAs were augmented with the lessons learned in previous IV&V phases, additional flight experiments, and continued monitoring of the operational experiences of military users of the selected COTS GPS receiver.

3.2 Tools

Software analysis tools are valuable in any IV&V effort because the tools can automate certain analysis tasks. Software tools are especially useful in the case of tailored-COTS because the majority of the software already exists when the project begins, so the tools can be used much earlier in the IV&V activity.

Many standard reverse engineering and software analysis tools are useful aids to IV&V. Among these are commercial tools intended to support maintenance of code [16] and various tools in an advanced state of research. For example, research tools exist that compute worst case execution time and that handle advanced programming constructions including: limited recursion, analytically complex loops with multiple exits, non-looping functions, function pointer calls, data pointers, non-terminating loops and functions, and multiple entry points [4]. Other useful tools produce diagrams to aid understanding and document the design [18]. Tools that compute cyclomatic complexity are also useful, particularly in support of the CARA, as cyclomatic complexity has been shown to be a reliable risk indicator [6].

Several static analysis tools are especially useful in identifying code that interacts extensively with new or changed code [4, 5, 15, and 21]. Set/use identification tools allow an analyst to rapidly assess the interactions from a data flow perspective. Flow chart generators and call trees provide a control flow perspective. Of course, these tools are also valuable during detailed analysis of the critical functions selected via the CARA.

Another class of tools that is particularly useful in the tailored-COTS environment is special purpose code audit tools. These are tools designed to automatically locate and assess particular patterns. For example, while on orbit, the Shuttle has line-of-sight visibility to more satellites than does a typical COTS GPS receiver user in atmospheric flight. Therefore, it was necessary to verify that all applicable tables and arrays are properly sized for the Shuttle environment. This task was well suited to a custom code analysis tool. Special purpose audit tools were also produced to rapidly locate additional instances

of problems identified from historical databases. Among these were tools to identify and check instances of function calls, to search for potential instances of division by zero, and to search for potential instances of indexing arrays beyond their limits.

3.3 Scenario Analysis

Software scenario analysis is a team problem solving technique that seeks to understand the behavior of a software system responding to various external events. A software scenario begins with an external event, and ends when the system resumes nominal cyclic operation or an error occurs. A similar team approach has been used to verify requirements for real time spacecraft systems [19] and relates to techniques for stepwise refinement and verification used in the Cleanroom approach [13].

Our approach to software scenario analysis can be summarized as the following sequence of activities:

- Using group-brainstorming techniques, a large number of potential scenarios are postulated. This is aided by both operational environment expertise and critical function expertise that analysts have gained in earlier phases of the IV&V project, particularly requirements analysis.
- Using a process similar to CARA, all scenarios are ranked based on criticality and risk.
- The primary IV&V analyst assigned to the critical function most involved in the scenario initiates analysis for each scenario. The analyst formally documents the control and data flows in a scenario analysis report.
- When flow passes to another critical function, analysis responsibility is transferred to the analyst with appropriate critical function expertise. This transfer is repeated until the scenario reaches a logical conclusion. Each analyst records his or her findings in the scenario analysis report.
- The lead analyst for the scenario presents the report at a peer review meeting and the entire scenario is discussed in detail. This step verifies the results and often suggests new scenarios and interactions with other critical functions.

Operational scenario analysis is frequently a valuable IV&V technique. But, it is particularly useful in the tailored-COTS environment because it is an efficient means to identify and evaluate the behavior of critical functions that are not changed but that are operationally affected by changes in other areas. In the case of the modified COTS GPS receiver, operational scenario analysis resulted in the identification of a number of subtle software issues. Additionally, operational scenario analysis was valuable in follow-on CARA updates and resulted in the inclusion of two new critical functions in the IV&V activity.

3.4 Model Checking

Model Checking is a formal verification technique in which assertions about a finite state machine process model are automatically tested [1– 3, 7 – 10, 15, 22, and 23]. Model checking is useful for a variety of verification approaches [20]. For example, it is useful as a means to assess liveness properties of the underlying finite state machine [8]. Model checking has also been demonstrated as means to automatically generate test cases [2].

The principal difficulty in model checking, from the analyst's perspective, is producing the model. It is necessary both to develop the model and to verify its equivalence to the system under consideration. Tailored-COTS can be an ideal candidate for model checking because the majority of the source code exists when IV&V begins. Consequently, it may be possible to automatically translate the source code into the modeling language, reducing labor and increasing the likelihood of an accurate model.

For the modified COTS GPS receiver IV&V project, model checking proved to be an extremely valuable adjunct to the scenario analysis process. For example, a critical portion of the COTS GPS receiver software (Receiver Manager) is implemented as a set of finite state machines. This critical function manages the five satellite tracking channels, which perform multiple tasks. The CARA suggested that this function was high in criticality and risk, and preliminary scenario analysis supported the CARA. Scenario analysis brainstorming revealed numerous scenarios with respect to Receiver Manager. Unfortunately, the complexity of the function would make manual analysis of all the scenarios prohibitively time consuming.

Since the source code was structured as a set of finite state machines, it was a straightforward task to translate the source code into the model checking language Promela [14] for use with the Spin model checker. Using Spin, it was possible to automatically check all of the Receiver Manager scenarios [1]. This allowed us to verify liveness properties of all of the possible configurations of the finite state machine. In particular, it identified a singular situation in which a receiver channel could be frozen in a certain state (a deadlock). Additionally, a byproduct of the model checking process is a scenario trace that shows how the deadlock state can be reached. This information greatly facilitated manual verification of the problem scenario.

3.5 Historical Databases

A major benefit of tailored-COTS is that it has an operational experience base. Insight into operational experience can greatly facilitate CARA and can also help to identify the unchanged but operationally affected code. Some of the sources of operational experience information are:

- User group databases. Since the COTS GPS receiver is a military product shared by all branches of the armed services, there is a joint program office that maintains valuable data. There are often USENET users' groups that can be significant sources of operational information.
- Vendor problem databases. These databases provide insight into both criticality and risk. In some cases, they may even contain useful information on previous tailoring of the COTS product. The COTS GPS receiver manufacturer maintains a problem database that was extremely beneficial to the IV&V effort.
- Test results. There should be a wealth of useful test results for any operational product. This information can augment the problem databases. However, because of its size, it should not be used as a primary reference. In the case of the modified COTS GPS receiver, several Shuttle missions gathered data using different versions of the receiver, including production prototypes.

4 Conclusions and Future Work

The Space Shuttle's modified COTS GPS receiver IV&V activity has demonstrated that COTS can be successfully tailored to operational environments for which it was not originally intended. The key difference between tailored-COTS IV&V and traditional IV&V is the need to identify and verify portions of the software that are not changed but that are operationally affected by the new environment. The techniques that proved most beneficial were a modified criticality analysis and risk assessment process, custom source code analysis tools, software scenario analysis, and model checking. Finally, historical databases were found extremely valuable sources of information.

There are significant opportunities for further research in the area of tailored-COTS IV&V. For example, tools that automatically extract finite state machine models from procedural language source code would facilitate model checking. There is also a need for tools to support the scenario analysis process and to support CARA.

5 Acknowledgements

The authors wish to acknowledge the support of AverStar, Inc. and the NASA IV&V facility in Fairmont, West Virginia. In particular, Prof. Jack Callahan of West Virginia University and Steve Husty of AverStar contributed extensively to the model checking activity. We also wish to acknowledge the members of the modified COTS GPS receiver IV&V team: John Bradbury, Reid Brockway, Don English, Larry Wiederholt, and David Wirkkala.

References

[1] Beims, M., and Callahan, J. *Independent validation and verification of firmware.* NASA 2nd Annual Workshop on Risk Management (WoRM 99). October 28-29,1999. Fairmont, WV.

[2] Callahan, J. *Model checking as a test case generator.* Work in Progress Presentation. Fall 1998. NASA Software IV&V Facility, Fairmont, WV.

[3] Clarke, E., and Gluch, D. *History of model checking.* SEI/CMU Site Visit Presentation. Winter 1998. NASA Software IV&V Facility, Fairmont, WV.

[4] Engblom, J. *Static properties of commercial embedded real-time programs, and their implication for worst-case execution time analysis. In:* Proceedings of the Fifth IEEE Real-Time Technology and Applications. 1999. pp 46 –55.

[5] Hayman, K. *An analysis of ordnance software using the MALPAS tools. In:* Proceedings of the Fifth IEEE on COMPASS '90, Systems Integrity, Software Safety, and Process Security. 1990. pp 86 - 94.

[6] Heimann, D. *CATS-an automated user interface for software development and testing. In:* IEEE Proceedings Annual Reliability and Maintainability Symposium. 1996. pp 163 - 166.

[7] Holzmann, G. *The spin model checker.* IEEE Trans. on Software Engineering, Vol. 23, No. 5. May 1997. pp 279 - 295.

[8] Holzmann, G., and Peled, D. *An improvement in formal verification.* Proceedings FORTE 1994 Conference, Bern, Switzerland.

[9] Husty, S. *An automated software maintenance process for the firmware using model checking techniques draft version.* Master of Science Project Report. University of West Virginia. 1999.

[10] Joseph, S. *Fault injection with model checking*, Ph.D. Thesis, University of West Virginia. December 1998.

[11] Kuehl, C. *A process direction for common avionics developments using commercial hardware and software components: The avionics systems engineering challenge. In:* Proceedings of the 16th Digital Avionics Systems Conference (DASC)., conference publication AIAA/IEEE Volume: 2. 1997. pp 6.4 -1- 6.4-9.

[12] McCaugherty, D. *The Criticality and Risk Assessment (CARA) method.* Workshop on Risk Management (WoRM) 98. 26 October 1998. http://research.ivv.nasa.gov/worm98/proceedings/mccaugherty.pdf

[13] Mills, H. *Stepwise refinement and verification in box-structured systems.* Cleanroom Software Engineering: A Reader. 1996. pp 169 - 197.

[14] Nagulakonda, V. *Creating models from source code.* Work in Progress Presentation. Winter 1998. NASA Software IV&V Facility, Fairmont, WV.

[15] Ogasawara, H., Aizawa, M., and Yamada, A., *Experiences with program static analysis. In:* Proceedings of the Fifth IEEE Software Metrics Symposium. 1998. pp 109 - 112.

[16] Oman, P., Novobilski, A., Rajlich, V., Harband, J., McCabe, T., Cross, J., Vanek, L., Davis, L., Gallagher, K., and Wilde, N. *Maintenance tools.* IEEE Software Volume: 7. 3 May 1990. pp 59 - 65.

[17] Perry, II. *The application of commercial processing technologies to the airborne military environment. In:* Proceedings of the 17th Digital Avionics Systems Conference (DASC)conference publication AIAA/IEEE/SAE Volume: 2. 1998. pp G35/1 - G35/8.

[18] Pierce, R., Ayache, S., Ward, R., Stevens, J., Clifton, H., and Galle, J. *Capturing and verifying performance requirements for hard real time systems. In:* Ada-Europe International Conference on Reliable Software Technologies conference publication Lecture Notes in Computer Science. 1997. pp 137 - 160.

[19] Prywes, N., Rehmet, P., Sokolsky, O., and Lee, I. *Retrospective exploration of safety properties in real-time concurrent systems. In:* Proceedings of the 16th Digital Avionics Systems Conference (DASC), conference publication AIAA/IEEE Volume: 1. 1997. pp 1.1 - 43 - 1.1 - 51.

[20] Schneider, F. *Verification and validation through model checking.* Jet Propulsion Laboratory. California Institute of Technology, Pasadena, CA. October 12, 1997.

[21] Thornley, J. *Static analysis and diversity in the software development process – Experiences with the use of SPARK. In:* Ada-Europe International Conference on Reliable Software Technologies, conference publication Lecture Notes in Computer Science. 1997. pp 266 - 277.

[22] Vijayakumar, S. *Use of historical data in software cost estimation.* Computing & Control Engineering Journal Volume: 8 3. June 1997. pp 113 - 119.

[23] Williams, C. *Spin modeling of CLCS redundancy management.* Work in Progress Presentation. Winter 1998. NASA Software IV&V Facility, Fairmont, WV.